AMENDMENTS TO THE CLAIMS

Claims 1-20, of which claims 1, 9, 17, 19 and 20 are independent, are currently pending in the application. This listing of claims replaces all previous versions and listings of claims. Please cancel claims 21 and 22.

1.   (Original)   A converting device, comprising: at least one IEEE 1394 interface for processing data in IEEE 1394 format by at least one of receiving and transmitting; at least one USB interface for processing data in USB format by at least one of receiving and transmitting; and processing logic coupled to the at least one IEEE 1394 and USB interfaces to convert data between IEEE 1394 protocol and USB protocol.

2.   (Original)   The converting device of claim 1, wherein the IEEE 1394 interface and USB interface comprise at least one protocol translation unit.

3.   (Original)   The converting device of claim 1, wherein the at least one IEEE 1394 interface comprises an IEEE 1394 socket.

4.   (Original)   The converting device of claim 3, wherein the IEEE 1394 interface comprises a PCI Link/Phy open host controller interface chip.

5.   (Original)   The converting device of claim 1, wherein the at least one USB interface comprises a USB socket.

6.   (Original)   The converting device of claim 5, wherein the at least one USB interface comprises a USB microcontroller chip with a high speed USB interface.

7.   (Original)   The converting device of claim 6, wherein the USB microcontroller chip comprises a general programmable interface.

8.   (Original)   The converting device of claim 1, wherein the processing logic comprises a processor executing instructions received from the at least one USB interface.

Page 2 of 6

9.   (Original)   A processing system for converting electronic data, comprising: a converter, comprising an IEEE 1394 interface and a converter USB interface; an IEEE 1394 device in communication with the at least one IEEE 1394 interface; and a computer system, wherein the computer system includes at least one system USB interface in communication with the at least one converter USB interface, and wherein the converter converts data units to facilitate transfer of the data units between the computer system and the IEEE 1394 device.

10.   (Original)   The device according to claim 9, wherein the IEEE 1394 device is a camcorder.

11.   (Original)   The device according to claim 10, wherein the IEEE 1394 device continuously streams data to the IEEE 1394 interface on the converter.

12.   (Original)   The device according to claim 9, further comprising: an application running on the system, wherein the application is configured to receive the data units.

13.   (Original)   A method for transferring data between a video application, executing in a computer system, and a digital video device, comprising: generating an IEEE 1394 command to exchange data with a digital video device, the IEEE 1394 command being generated in the application according to IEEE 1394 protocol; converting the IEEE 1394 command to a USB command in accordance with the USB protocol; transmitting the USB command over a USB connection external to the computer system; receiving the USB command in a converter device; converting the USB command to the IEEE 1394 command; and transmitting the IEEE 1394 command to the digital video device.

14.   (Original)   The method as recited in claim 13, wherein converting the IEEE 1394 command to the USB command comprises a 1394-USB tunnel driver that receives a IEEE 1394 bus input-output request, creates data packets with OHCI-compatible PCI accesses and transmits them to a USB stack.

15.   (Original)   The method as recited in claim 13, wherein converting the IEEE 1394 command to the USB command comprises, a USB client device driver that receives data packets from a IEEE 1394 stack, passes data packets to a system driver component, and transmits them to the converter device.

16.   (Original)   The method as recited in claim 13, wherein converting the IEEE 1394 command to the USB command comprises first converting the IEEE 1394 command to an intermediate protocol.

17.   (Original)   A method for transferring data between a video application, executing in a computer system, and a digital video device, comprising: generating an IEEE 1394 command to exchange data with the video application running on the computer system, the IEEE 1394 command being generated in the digital video device according to IEEE 1394 protocol; transmitting the IEEE 1394 command over a IEEE 1394 to a converter device; receiving the IEEE 1394 command in the converter device; converting the IEEE 1394 command to a USB command in accordance with the USB protocol; transmitting the USB command over a USB connection; and receiving the USB command in the computer system.

18.   (Original)   The method as recited in claim 17, wherein converting the IEEE 1394 command to the USB command comprises first converting the IEEE 1394 command to an intermediate protocol.

19.   (Original)   A method for transferring data between a computer system executing a video application and a digital video device, comprising: receiving a USB data packet in USB protocol from a USB interface, wherein the USB data packet comprises a header portion and a data portion; storing the USB data packet in a buffer; retrieving the header portion of the USB data packet from the buffer; analyzing the header portion of the USB data packet to determine a command parameter; storing the command parameter into an IEEE-1394 compliant OHCI register; retrieving the data portion of the USB data packet from

Page 4 of 6

the buffer; assembling an IEEE 1394 command in IEEE 1394 protocol from the command parameter and the data portion of the USB data packet; transmitting the IEEE 1394 command in IEEE 1394 protocol to the digital video device over an IEEE 1394 interface.

20.   (Original)   A method for transferring data between a computer system executing a video application and a digital video device, comprising: receiving an IEEE 1394 response from a digital video device in IEEE protocol from an IEEE 1394 interface; analyzing the IEEE 1394 response to determine a response command parameter; adding a response header portion to the response command parameter to form a response USB data packet; and transmitting the response USB data packet in USB protocol over a USB interface.

21-22. (Canceled)